INTELLIGENT FILESYSTEM GUARD

Jan Hykel, Master Degree Programme (5)
Dept. of Intelligent Systems, FIT, BUT
E-mail: xhykel00@stud.fit.vutbr.cz

Supervised by: Ing. Aleš Smrčka

ABSTRACT

Intelligent Filesystem Guard is a tool that monitors information about changes in the files and directories. This program can be used either for the detection of changes in the important files (Intrusion Detection System guarding the data integrity - viruses, Trojan horses) or for guarding the user data. A large emphasis is put on monitoring files. One of the functions of this system is to tell what happened with the file according to a user query. The system is able to warn of whichever change, that means modification, creation, erasure or move.

1 ÚVOD

Při práci s počítačem prakticky neustále využíváme služeb souborového systému jakožto prostředku abstrakce přístupu k diskové paměti. Vytváříme soubory, modifikujeme jejich obsah, můžeme je přesouvat na jiné místo v souborovém systému. Pokud již soubor není potřeba, lze jej jednou provždy vymazat. Každý soubor prochází tedy jednotlivými životními etapami (životním cyklem). Cílem níže popsaného projektu je vytvořit komplexní systém, který dokáže sledovat jakékoli změny probíhající v souborovém systému, tyto změny uchovávat, vyhodnotit a reagovat na ně.

2 ŽIVOTNÍ CYKLUS SOUBORU

Souborový systém je navržen jako dynamická datová struktura, která obsahuje informace o souborech. Nezajímá nás nyní obsah, ale další údaje jako vlastník, přístupová práva, velikost, čas přístupu a další. Tyto údaje popisují stav souboru, který je platný nyní. Souborový systém nám může poskytnout informace o souborech, ale jen ty, které platí teď. Nemůže poskytnout informace o stavu, který byl platný například před pěti dny. Soubory a adresáře jsou ale dynamické entity, jejichž obsah a vlastnosti se mění v čase. Je zřejmé, že každý soubor má svou historii. Jedním z cílů projektu je i uchovávání těchto údajů. Oprávnění uživatelé pak smí v databázi historii vyhledávat.

3 NÁVRH SYSTÉMU

Systém se skládá ze čtyř hlavních částí, které vzájemně spolupracují. První část řeší abstrakci pro přístup k souborům, druhá část provádí indexovací algoritmy nad vstupní množinou souborů, třetí část provádí analýzu konfiguračních souborů a konečně poslední část řeší ukládání a načítání datových struktur. Je použit objektově orientovaný návrh za pomoci jazyka UML [1].

3.1 PŘÍSTUP K SOUBOROVÉMU SYSTÉMU

Přístup k souborovému systému je řešen mezivrstvou mezi výkonnou částí a souborovým systémem. Při požadavku provede mezivrstva načtení informací o souboru a v upravené podobě zašle potřebné údaje výkonné části, která s nimi dále pracuje.

Definujme nyní množinu souhrnných informací o všech souborech (adresářích) $F = \{f_1, f_2, ..., f_m\}$. Dále definujme množinu souhrnných údajů o všech souborech, které existují v souborovém systému jako $\varphi \subseteq F$. Tato množina představuje *aktuální stav* a byla získána pomocí rekurzivního průchodu (preorder) adresářovým stromem souborového systému.

3.2 INDEXOVACÍ ALGORITMY

Vstupem algoritmu indexace je množina φ a dále množina $\sigma \subseteq F$, která představuje *minulý stav*. Během indexování se provádí výpočet následujících množin: nejprve je to množina *možných přidaných* souborů $\alpha = \varphi \setminus \sigma$ a množina *možných odebraných* souborů $\beta = \sigma \setminus \varphi$. Definujme nyní funkci $hash : F \to \mathbb{N}$. Množina *přesunutých* souborů je pak určena vztahem:

$$\delta = \{x \mid x \in \alpha \wedge \exists y \in \beta : hash(x) = hash(y)\}$$

Dále se vypočte množina *přidaných* souborů $\gamma = \alpha \setminus \delta$ a nakonec množina *odebraných* souborů:

$$\varepsilon = \beta \setminus \{x \mid x \in \beta \wedge \exists y \in \alpha : hash(x) = hash(y)\}$$

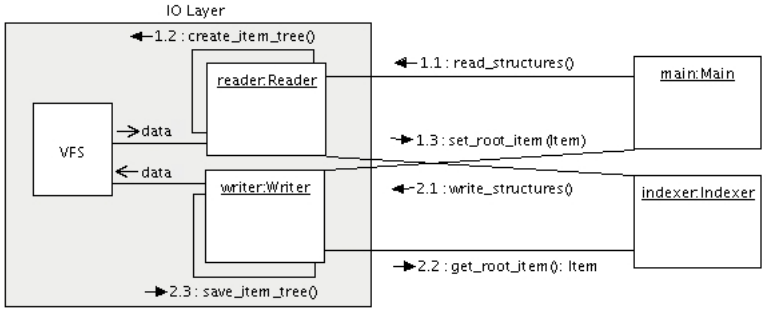
Množiny δ, γ a ε jsou po dalším zpracování uchovány jako součást nového stavu souborového systému.

3.3 ANALÝZA KONFIGURAČNÍHO SOUBORU

Navržená forma konfiguračních pravidel byla inspirována notací použitou v hodně rozšířeném systému detekce průniků TripWire [2]. Každé pravidlo je trojice:

```
jméno_objektu -> maska_vlastností -> akce;
```

Jméno_objektu představuje absolutní cestu k souboru nebo adresáři definovanou regulárním výrazem. *Maska_vlastností* specifikuje, které vlastnosti nad daným objektem kontrolovat nebo ignorovat. Pokud je pravidlo splněno, je vyvolána *akce*. Tato akce musí obsahovat jediný příkaz (nebo jméno skriptu). K jednomu souboru nebo adresáři se může vztahovat více pravidel. Maska vlastností je prostředek pro popsání podmínek, při jejichž splnění



Obrázek 1: V/V vrstva (collaboration diagram)

dojde k provedení nějaké akce. Maska vlastností se skládá ze sekvence jednoznakových symbolů. Každý jednoznakový symbol určuje vlastnost, jejíž kontrolu aktivujeme.

3.4 MEZIVRSTVA PRO V/V

Tato vrstva zajišťuje načítání a ukládání dat. Data jsou uložena v datovém úložišti, které je implementované pomocí klasických sekvenčních souborů (viz obr. 1). Je možné dopsat podporu pro jiný typ úložiště.

4 IMPLEMENTACE

Implementace projektu probíhá v jazyce C++ za použití standardní knihovny šablon (STL). Projekt je publikován na freshmeat.net (http://freshmeat.net/projects/fkeeper) a je šířen pod licencí GNU GPL [4].

5 ZÁVĚR

Existuje mnoho nástrojů pro kontrolu souborového systému podobného typu, jako je tento projekt. Většina z nich je však určena pouze pro kontrolu integrity souborů (např. TripWire [2], AIDE [3]), nebo se jedná o specializované komerční nástroje pro detekci průniků, virů a trojských koní. Navíc žádný (ani komerční) programový nástroj není schopen sledovat životní cyklus souboru. Popsaný systém by měl umožnit funkční spojení všech těchto samostatných nástrojů do jednoho komplexního celku. Jeho použití je univerzální a díky POSIX standardizaci je přenositelný na mnoho různých operačních systémů.

REFERENCE

[1] Unified Modeling Language, URL: http://www.uml.org/

[2] Nástroj pro kontrolu integrity systému TripWire, URL: http://www.tripwire.com/

[3] Advanced Intrusion Detection Environment, URL: http://aide.sourceforge.net/

[4] GNU General Public License, URL: http://www.gnu.org/licenses/gpl.html